Amendments to the Claims

The listing of claims will replace all prior versions, and listings of claims in the application:

1. *cancelled*

2. (*currently amended*) A method for prequeuing [of] files predicted to be desired by a user, comprising:

defining a ~~restrictive criteria to select~~ search criterion that, when applied, selects a list of files, such that each file is interrelated to other files in accordance with the context of the search criterion;

defining a sort criterion that, when applied, sorts the selected files in a user-specified order;

receiving a list representing the selected files in the sorted order;

automatically transferring files on the list into a local cache, in anticipation of a user selection thereof, files already transferred to the local cache having a shorter delay for review than those which have not been previously transferred to the local cache, an order of file transfer being responsive to ~~a prediction of user review requirements~~ the sort criterion, the ~~prediction~~ order of file transfer being further responsive to any change in a user deviation ~~from the predicted~~ in an order of reviewing the selected files; and

receiving a starting point within the list of files, for file review, from the user, such that predicted latencies for sequential file review from any given starting point are reduced.

3. (*currently amended*) The method according to claim [1] 31, wherein said method is executed on the client by a browser application.

4. (*currently amended*) The method according to claim [1] 31, wherein said method is executed on the client by a browser plug-in or extension.

5. (*currently amended*) The method according to claim [1] 31, further comprising:

~~the step of~~ cost accounting for downloading of each ~~record~~ object.

6. (*currently amended*) The method according to claim [1] 31, further comprising ~~the steps of~~:

communicating through a network to a server hosting the ~~records~~ objects; and

presenting [a] the list ~~of records~~ to [a] the user prior to receiving a selection of ~~a record~~ an object from the user.

7. (*currently amended*) The method according to claim [1] 31, further comprising ~~the steps of~~:

accounting for a downloaded ~~record~~ object; and

limiting said downloading based on a predetermined parameter.

8. (*previously presented*) The method according to claim 2, wherein predicted latencies are minimized.

9. (*previously presented*) The method according to claim 2, wherein the transferring of files is executed to reduce predicted latencies or increase a throughput of the connection between a source of the files being transferred and the local cache.

10. (*currently amended*) The method according to claim 2, wherein the transferring of files is executed to reduce predicted latencies or implement an apparent strategy for review of ~~records~~ files by the user.

11. (*currently amended*) The method according to claim 2, wherein the transferring of files is executed to reduce predicted latencies or a cost of the ~~record~~ file downloads.

12. (*previously presented*) The method according to claim 2, wherein the transferring of files is executed to reduce predicted latencies or a cost of on-line time.

13. (*previously presented*) The method according to claim 2, wherein the transferring of files is executed to reduce predicted latencies or increase a value of the user's time.

14. (*currently amended*) The method according to claim 2, wherein the transferring of files is executed to reduce predicted latencies or a burden on ~~the~~ a server hosting the files.

15. (*currently amended*) A browser, comprising:

a first interface for defining ~~a restrictive criteria to define~~ a search criterion that, when applied, produces a list identifying a set of objects, wherein the objects are interrelated to each other in accordance with the context of the search criterion;

a second interface for defining a sort criterion that, when applied, sorts the identified objects in a user-specified order;

logical elements for enabling automatic transferring of an object identified in the list into a cache local to a user, in advance of an actual selection of an object by the user, objects already transferred to the local cache having a lower latency than those which have not been previously transferred to the local cache, an order of object transfer being responsive to ~~a prediction of user requirements~~ the sort criterion, the logical elements being adaptive to a user deviation ~~from the predicted~~ in an order of reviewing the identified objects; and

a ~~second~~ third interface for receiving a selection of an object as a starting point within the list of objects, such that predicted latencies for sequential object browsing from any given starting point are reduced.

16. (*currently amended*) The browser according to claim 15, further comprising:

an accounting system for accounting for downloading of each object.

17. (*previously presented*) The browser according to claim 15, wherein predicted latencies are minimized.

18. (*previously presented*) The browser according to claim 15, wherein the transferring of objects is executed to reduce predicted latencies or increase a throughput of the connection between a source of the objects being transferred and the local cache.

19. (*previously presented*) The browser according to claim 15, wherein the transferring of objects is executed to reduce predicted latencies or a cost associated with the object transfers.

20. (*previously presented*) The browser according to claim 15, wherein the transferring of objects is executed to reduce predicted latencies or increase a value of the user's time.

21. (*currently amended*) A method for transferring files for sequential review, comprising:

accessing a ~~restrictive criteria to select~~ search criterion that, when applied, selects a list of files, such that each file is being interrelated to the other files in accordance with the context of the search criterion;

determining an order of file transfer based on a sort criterion specified by a user;

queuing the files on the list according to the order of file transfer; and

transferring automatically the queued files in a sequential order into a local cache for sequential review at a client.

22. (*previously presented*) A method according to claim 21, wherein said transferring comprises:

receiving a user-specified quantity of files to be transferred; and

transferring the user-specified quantity of files into the local cache.

23. (*previously presented*) The method according to claim 21, further comprising:

receiving a format change during said transferring; and

transferring subsequent files on the list according to the format change.

24. (*previously presented*) The method according to claim 21, further comprising:

receiving a revised sort criterion from the client;

determining a revised order of file transfer based on the revised sort criterion; and

queuing the files on the list according to the revised order of file transfer.

25. (*previously presented*) The method according to claim 21, further comprising:

sending the list to the client; and

receiving, from the client, a first selection of a file from the list, wherein said queuing comprises queuing the first selection as the first file and queuing subsequent files according to the order of file transfer.

26. (*previously presented*) The method according to claim 21, further comprising:

receiving a request to cancel an item from the list; and

removing the item from at least one of the list or the queued files.

27. (*currently amended*) The method according to claim 21, further comprising:

receiving a non-sequential request for an item on the list, wherein said queuing comprises queuing the requested item as the first file and queuing subsequent files according to the order of file transfer.

28. (*previously presented*) The method according to claim 21, further comprising:

receiving a user-specified image resolution format or a user-specified file format for the files; and

processing the files to comply with the image resolution format or the file format prior to executing said transferring.

29. (*currently amended*) The method according to claim 21, further comprising:

receiving a user-defined parameter to establish a size of a queue for holding the queued files.

30. (*currently amended*) The method according to claim 29, further comprising:

transferring, into the local cache, all files held in the queue at once.

31. (*new*) A method for receiving objects at a client from a remote processing system, comprising:

accessing, at the client, a search request that, when executed, identifies a plurality of objects, each object being interrelated to the other objects in accordance with the context of the search criterion;

accessing, at the client, a sort criterion that, when executed, sorts the identified objects in a user-specified order;

receiving, at the client, a list representing the identified objects in the sorted order; and

receiving, in a cache at the client, one or more of the identified objects in the sorted order prior to receiving, at the client, a request to present at least one object from the list to a user.

32. (*new*) The method according to claim 31, further comprising:

sending the search request to the remote processing system that assembles the identified objects in the sorted order.

33. (*new*) The method according to claim 32, further comprising:

accessing a user-specified queue parameter that, when applied, sets a size of a queue that holds the assembled objects prior to transfer to the client.

34. (*new*) The method according to claim 31, further comprising:

accessing a user-specified cache parameter that, when applied, sets a size of the cache that holds the one or more objects received in the cache.

35. (*new*) The method according to claim 31, further comprising:

accessing a user-specified format parameter that, when applied, sets a format for the identified objects.

36. (*new*) The method according to claim 35, further comprising:

utilizing the format on at least one of the identified objects to process the at least one object for transfer to the cache or for presentation from the cache;

accessing a second user-specified format parameter that, when applied, sets a second format for the identified objects; and

utilizing the second format to process at least one of the identified objects that are assembled for transfer to the cache after the application of the second user-specified format parameter or to process at least one of the identified objects that are presented from the cache after the application of the second user-specified format parameter.

37. (*new*) A system for transferring a plurality of objects for sequential browsing, comprising:

a client having a first interface that accesses a search criterion to define a list identifying a set of objects, wherein the objects are interrelated to each other in accordance with the context of the search criterion, wherein the client is coupled to a cache that includes an object from the list, and wherein the client is coupled to a display that enables a user to browse an object from the list; and

an object retrieval server that queues objects from the list in a user specified order, wherein the server automatically transfers the queued objects according to the user specified order to the cache in advance of an actual selection of an object for browsing by the user, and wherein the user specified order is responsive to a change in a user deviation in an order of reviewing the objects.

38. (*new*) The system of claim 37, wherein the client includes a second interface that accesses a sort criterion that, when executed, queues the objects in said user specified order.

39. (*new*) The system of claim 37, wherein the client includes a second interface that receives a selection of an object as a starting point within the list, such that predicted latencies for sequential object browsing from any given starting point are reduced.

40. (*new*) A computer program product comprising a computer useable medium having computer readable program code functions embedded in said medium for causing a computer to transfer files for sequential review, comprising:

a first computer readable program code function that causes the computer to access a search criterion that, when applied, selects a list of files, wherein each file is interrelated to the other files in accordance with the search criterion;

a second computer readable program code function that causes the computer to determine a user specified order of file transfer based on a sort criterion; and

a third computer readable program code function that causes the computer to transfer automatically the files according to the order of file transfer into a local cache for review at a client.

41. (*new*) The computer program product of claim 40, further comprising:

a fourth computer readable program code function that causes the computer to queue the files on the list according to the order of file transfer, wherein the third computer readable program code function automatically transfers the queued files into the local cache.

42. (*new*) The computer program product of claim 40, further comprising:

a fourth computer readable program code function that causes the computer to transfer a user-specified quantity of files into the local cache.